

A KNOWLEDGE-BASED METHOD FOR GENERATING SUMMARIES OF SPATIAL MOVEMENT IN GEOGRAPHIC AREAS

MARTIN MOLINA

*Department of Artificial Intelligence, Technical University of Madrid,
Campus de Montegancedo S/N 28660 Madrid, Spain,
martin.molina@upm.es*

AMANDA STENT

*AT&T Labs – Research,
180 Park Avenue, Florham Park, New Jersey 07932, USA
stent@research.att.com*

In this article we describe a method for automatically generating text summaries of data corresponding to traces of spatial movement in geographical areas. The method can help humans to understand large data streams, such as the amounts of GPS data recorded by a variety of sensors in mobile phones, cars, etc. We describe the knowledge representations we designed for our method and the main components of our method for generating the summaries: a discourse planner, an abstraction module and a text generator. We also present evaluation results that show the ability of our method to generate certain types of geospatial and temporal descriptions.

Keywords: Spatial abstraction; temporal abstraction; data-to-text system; natural language generation.

1. Introduction

It is now possible to collect, automatically and in real time, large quantities of geospatial sensor data (tracking features of individuals over time and space). For example, we can collect near-real-time measurements of the movements of individuals (see Figure 1) via GPS (Global Positioning Satellite) sensors embedded in cell phones and cars, of the movements of ships via AIS (Automatic Identification System), of the movements of data via network traffic monitoring, or of the movements of inventory via RFID (Radio Frequency Identification) tags. Therefore, it is important to have tools that help humans interpret and analyze these large datasets quickly, to facilitate tasks such as fleet monitoring, transportation analysis, mission data analysis (military or scientific missions), wildlife tracking, trip planning, and so on.

In the field of Artificial Intelligence, systems have been developed that summarize large data sets, to help humans interpret and analyze trends in the data. For example, data-to-text systems have recently been developed that generate textual summaries of numeric and other non-linguistic data to help people understand weather forecasts,²⁶

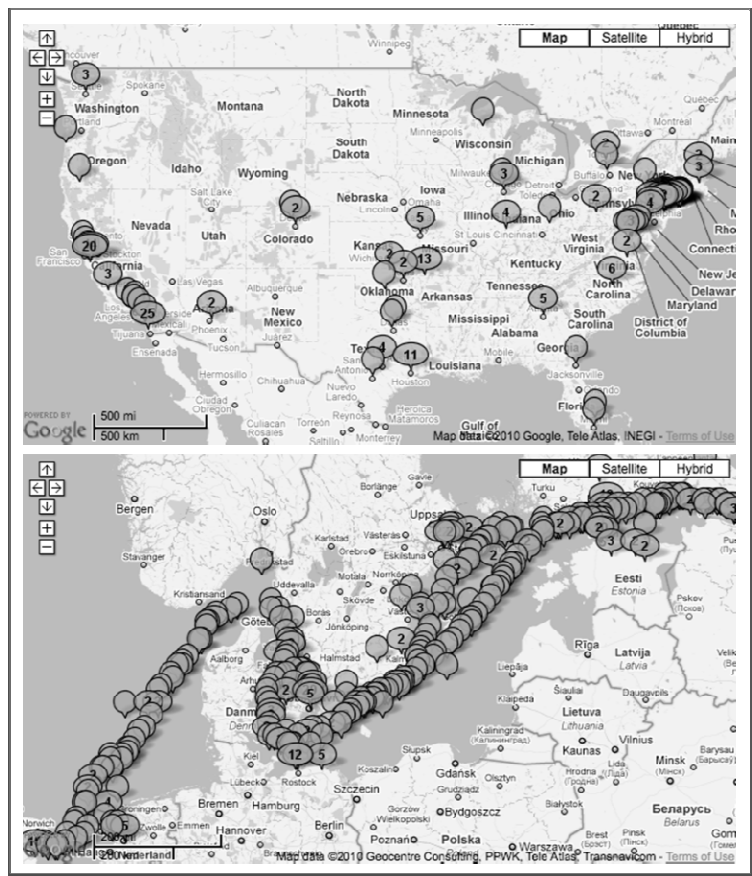


Fig. 1. The image at the top shows the geographic distribution of credit card purchases made by one individual over an 18-month period from 2008-2009. The image at the bottom shows a partial plot of movement of the Celebrity Constellation cruise ship over the period July-August 2009.

engineering data³⁷ or medical records.¹⁰ Other systems have been developed that produce multimedia summaries (for example, in hydrology¹⁸). However, these solutions are domain-dependent; they cannot be easily reused for new tasks and data sets. Some researchers have tried to make more general-purpose tools. For example, there are systems for summarizing temporal data,²⁹ and for summarizing geospatial data.¹² However, there are currently no methods that integrate both dimensions to generate summaries of spatial movement in geographic areas.

To address this need, we have developed a research agenda with the goal of building reusable tools for automatic generation of descriptions that summarize geospatial sensor data tracking features of individuals over time and space. Our work is initially focused on the generation of textual summaries intended to accompany multimedia presentations. The work presented in this article updates and extends preliminary reported work.^{19,31} The contributions of our work so far include the analysis of human-authored summaries

of geospatial/temporal data and the design of a method for generating this type of summaries. We implemented a prototype of our summarization method and performed an evaluation that showed its ability to generate certain types of geospatial and temporal descriptions.

Since the geospatial/temporal domain is present in many different types of tasks, we designed our approach to be reusable (wholly or partially) for different tasks and also to be part of other more complex domains. Our design follows a modular structure with knowledge representations that can be reused and extended with domain-dependent summarization strategies to build more complex summarization systems (for example, systems that make trip recommendations, or that analyze social networks).

The rest of this article describes the details of our method. First, we describe the summarization task. Then, we describe the main components of our method: a discourse planner, an abstraction module and a text generator. We focus on the details of our representations for (a) geospatial and temporal information and (b) discourse structures of summaries. Finally, we present an evaluation of our method and a comparison with related work.

2. Analysis of the Summarization Task

Our long-term research goal is to create general tools for systems that can automatically generate interactive multimedia presentations that summarize geospatial and temporal data. This task involves several subtasks, including data abstraction, discourse planning, text generation, and multimedia presentation generation. In this paper we focus on the discourse planning, data abstraction and text generation tasks.

A trip to Costa Rica and Panama

The trip started on January 2nd 2003 and lasted for two months. San Jose was the initial and final place of the trip. During the first 5 weeks, we stayed in Puntarenas most of the time. The rest of the time, we were travelling to different places of Costa Rica and Panama.

During the time we were in Puntarenas, we visited some nearby places on the Pacific Coast. For example, we went every weekend to San Lucas during the morning and we went three nights to Bajamar. We also visited Tarcoles and Puerto Caldera some days during the morning.

Fig. 2. Partial example of a geospatial/temporal summary.

The input to the summarization task is a quantitative dataset (non linguistic) that includes traces corresponding to the monitored individual movements of an object (e.g. a person, an animal or a vehicle) that record the object's varying spatial locations as it moves across a geographical area over time. We assume continuous point movements in geographic space, as it is defined by Yattaw.³⁵ These traces include date/time, latitude and longitude (as well as, potentially, other information such as wind speed, temperature, and companions) sampled periodically over a certain interval of time (for example, a number of days, months or years). The output from the method is a short natural language description (no more than 200 words) that summarizes the behavior of the individual in

geospatial and temporal terms. The summary is intended to accompany one or more visualizations of the input data (e.g. a map, a timeline). Figure 2 shows a representative example of our desired output.

We conducted an analysis of human-authored summaries of geospatial/temporal data available on the web (for example, *mapmyride.com*, *nytimes.com*, *travelblogs.com* and *tripdiary.com*). Table 1 shows a list of strategies we identified in our analysis of naturally occurring geospatial/temporal summaries. In this figure, we follow the categories used by Polifroni *et al.*²² (singleton, count, enumerate and majority). We identified several challenges for the design of a method for generating this type of summary, including:

- *Geospatial aggregation.* The summaries include names corresponding to geospatial abstractions — geographical units of different scale, e.g. countries, states, regions, cities. For example, the names *Puntarenas* and *Costa Rica* are really abstractions over sets of latitude/longitude pairs. Generating these descriptions requires geographical knowledge about names of places at different geographic scales (locations, regions, states, countries, etc.), as well as knowledge about how to cluster input data into the right units of geographical abstraction for realization using natural language.
- *Temporal aggregation.* The summaries also include temporal abstractions. For example, *two months* and *the first 5 weeks* are abstractions over time intervals which are themselves abstractions over sets of time points in the input data. This requires knowledge about time and calendars at different temporal scales (hour, day, month, year, etc.), as well as knowledge about temporal clustering that is amenable to realization in natural language.

Table 1. Examples of linguistic strategies to describe datasets with geospatial and temporal information.

Strategies	Examples
Singleton (place)	All the time the car was in Madrid
Singleton (place – date)	The car went to Madrid on Saturday
Singleton-out (place)	The car never went to Madrid
Count (places)	The car went to 5 towns
Count (duration)	The car was in Madrid for 3 weeks
Count (date)	The car was in Madrid 4 Sundays
Count (times)	The car was in Madrid 5 times
Count (periodicity)	The car went to Madrid 2 days a week
Enumerate (chronologic)	First, the car went to Madrid, then to Seville and finally to Valencia
Enumerate (by duration)	The car was in Madrid for 3 days, Seville for 2 days and Valencia for 1 day
Enumerate (by times)	The car was in Madrid 6 times, Seville 3 times and Valencia 2 times
Enumerate (random)	He went to Madrid, Seville and Valencia (order is not important)
Enumerate (examples)	For example, he went to Madrid and Seville (partial list)
Enumerate and count	The car went to Madrid, Seville and 3 other cities
Majority (time – places)	Most of the time the car was in Madrid, Seville and Valencia
Majority (places – duration)	Most of the places were visited in one hour or less
Majority (places – date)	Most of the places were visited during the weekend
Majority (trips – places)	In the majority of the trips, the car visited Madrid and Seville
Majority (trips – duration)	The majority of the trips lasted less than one month
Majority (trips – date)	The majority of the trips were in the summer

- *Temporal patterns.* Another characteristic of this type of summary is the description of temporal periods that summarize sets of events. For example, *every weekend to San Lucas during the morning* identifies a periodic visit to a particular place. The generation of descriptions about temporal patterns requires knowledge about time and calendars together with knowledge about description strategies.
- *Variable level of abstraction.* The level of abstraction in a summary is variable. The sentences of the text that summarize the geospatial and temporal data are presented at a level of abstraction that is dynamically chosen according to the input data and length restrictions on the output summary.
- *Variable structure of discourse.* The discourse structure of these summaries is not rigid, but is dynamically constructed according to characteristics of the input data and the desired length of the summary. Normally, a summary is structured so that it proceeds from a high level of geospatial and temporal abstraction to more specific information. However, the order of summary statement and supporting detail is variable, as is the arrangement of supporting detail. For example, the summary can include a complete chronological description of stops along a route, a few illustrative examples of places visited, a list of places ordered by duration of stay, etc.

3. The Method for Generating Summaries

We designed a knowledge-based method for geospatial/temporal summarization. Our method involves two main steps, *discourse planning* and *presentation generation*:

1. In the first step we generate the discourse plan. This is done in a loop involving two components: *discourse planning* and *abstraction* (see the top part of Figure 3). At each iteration, the *discourse planner* generates abstraction goals and the *abstraction component* returns the abstractions corresponding to these goals. This process is directed by a hierarchical search over possible discourse patterns, taken from a knowledge base, that are progressively refined until the final discourse plan is constructed.
2. In the second step we generate the output multimedia presentation. This is done in a loop involving *presentation planning* and *text generation* components (see the bottom part of Figure 3). At each iteration, the *presentation planner* produces a partial discourse and the *text generator* returns the generated text corresponding to the partial discourse.

In the following sections, we describe these steps in more detail. Because the focus of this article is the production of text summaries, we pay special attention to the components for discourse planning and text generation.

3.1. The representation of the discourse plan

The discourse plan represents the content of the presentation (what to present) without details of the presentation (how to present it). It consists of three types of information: information about the rhetorical structure of the discourse as a whole; information about

the objects mentioned in the discourse, used to ensure discourse coherence; and information about the semantics of the propositions in the discourse. We will now describe each of these parts of the representation with reference to the following example: “the ambulance went to NYC five weekends; for example, the ambulance went to Central Park on Saturday the 14th in the morning and to Brooklyn on Sunday the 22nd in the afternoon”.

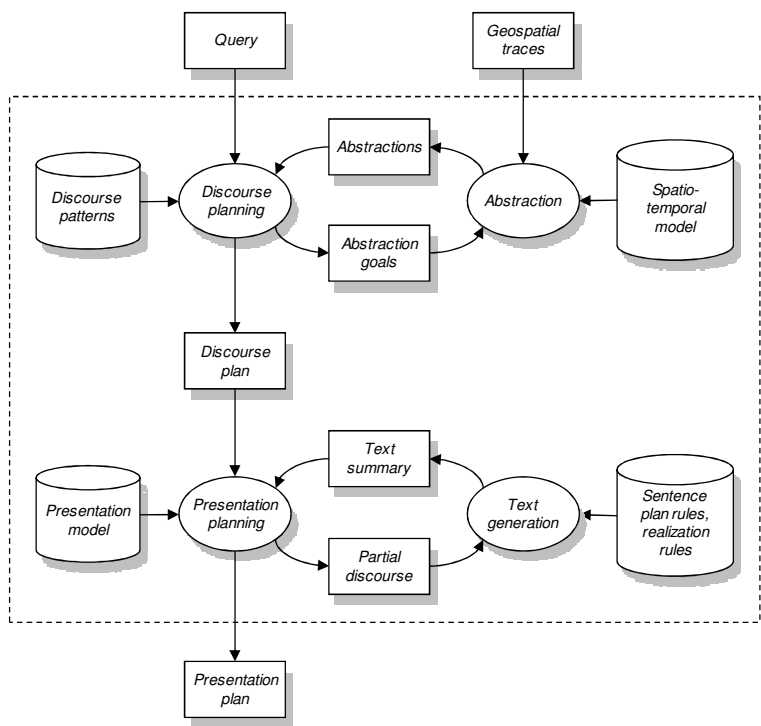


Fig. 3. Components of our knowledge-based method for generating descriptions that summarize spatial movement in geographic areas (rectangles represent input/output data, ellipses represent tasks or subtasks, and cylinders represent knowledge bases).

The rhetorical structure of the discourse is represented as a tree whose leaves are a set of propositions and whose internal nodes are rhetorical relations.¹⁴ The discourse tree for our example is shown in Figure 4. In our method, we currently use 14 rhetorical relations: *temporal_sequence*, *temporal_after*, *temporal_before*, *temporal_same_time*, *elaboration_general_specific*, *elaboration_part_whole*, *elaboration_process_step*, *elaboration_additional*, *sequence*, *list*, *exemplify*, *concession*, *reason*, and *restatement/summary*.

Each node in the discourse tree is represented using the predicate *node(x, y)* where *x* is an identifier (integer number) and *y* is the content. The content is one of the following:

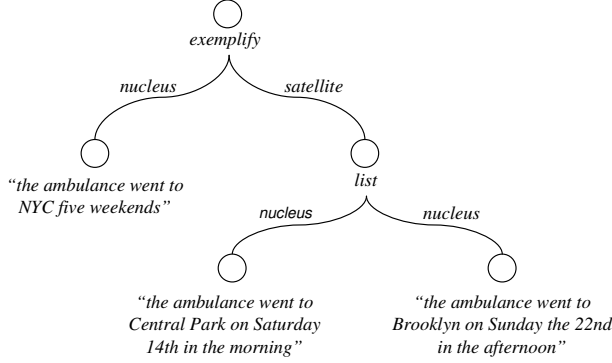


Fig. 4. Tree representation for a discourse plan.

- If the node is tree-internal, the content is represented using the predicate *relation*(*x*, *nucleus*(*y*), *satellite*(*z*)) where *x* is the name of a relation, *y* and *z* are ordered sets of node identifiers. For example *relation*(*list*, *nucleus*({2,3}), *satellite*({ })).
- If the node is a leaf, the content is represented using the predicate *proposition*(*x*) where *x* is a set of elements representing the semantics of a single proposition. For example, *the ambulance went to Central Park on Saturday the 14th in the morning* is represented as *proposition*({*event*(*x*₂), *a0*(*x*₂, *o*₁), *pred*(*x*₂, *go*), *place*(*x*₂, *o*₃), *date*(*x*₂, XXXX-14-WXX-06TMO)}).

Along with the discourse tree, we store representations of the set of objects mentioned in the discourse. This permits us to track mentions of objects across propositions to ensure discourse coherence and perform generation of referring expressions.¹⁷ To describe an object, we use the predicate *object*(*x*, *y*) where *x* is an identifier and *y* is a set of attributes that characterize the object. For example *object*(*o*₄, {*lex*(*Brooklyn*)}) could be realized as *the borough of Brooklyn*, *Brooklyn*, or *there*.

<p>OBJECTS:</p> <p><i>object</i>(<i>o</i>₁, {<i>lex</i>(‘the ambulance’), <i>type</i>(<i>agent</i>)})</p> <p><i>object</i>(<i>o</i>₂, {<i>lex</i>(<i>NYC</i>), <i>type</i>(<i>city</i>)})</p> <p><i>object</i>(<i>o</i>₃, {<i>lex</i>(‘Central Park’), <i>type</i>(<i>address</i>), <i>latitude</i>(40.7643), <i>longitude</i>(-73.9730)})</p> <p><i>object</i>(<i>o</i>₄, {<i>lex</i>(<i>Brooklyn</i>), <i>type</i>(<i>address</i>), <i>latitude</i>(40.6508), <i>longitude</i>(-73.9496)})</p> <p>DISCOURSE:</p> <p><i>node</i>(1, <i>proposition</i>({<i>event</i>(<i>x</i>₁), <i>a0</i>(<i>x</i>₁, <i>o</i>₁), <i>pred</i>(<i>x</i>₁, <i>go</i>), <i>place</i>(<i>x</i>₁, <i>o</i>₂), <i>dates</i>(<i>x</i>₁, 5, XXXX-XX-WE)})))</p> <p><i>node</i>(2, <i>proposition</i>({<i>event</i>(<i>x</i>₂), <i>a0</i>(<i>x</i>₂, <i>o</i>₁), <i>pred</i>(<i>x</i>₂, <i>go</i>), <i>place</i>(<i>x</i>₂, <i>o</i>₃), <i>date</i>(<i>x</i>₂, XXXX-14-WXX-06TMO)})))</p> <p><i>node</i>(3, <i>proposition</i>({<i>event</i>(<i>x</i>₃), <i>a0</i>(<i>x</i>₃, <i>o</i>₁), <i>pred</i>(<i>x</i>₃, <i>go</i>), <i>place</i>(<i>x</i>₃, <i>o</i>₄), <i>date</i>(<i>x</i>₃, XXXX-22-WXX-07TAF)})))</p> <p><i>node</i>(4, <i>relation</i>(<i>list</i>, <i>nucleus</i>({2,3}), <i>satellite</i>({ })))</p> <p><i>node</i>(5, <i>relation</i>(<i>exemplify</i>, <i>nucleus</i>({1}), <i>satellite</i>({4})))</p>
--

Fig. 5. Example of representation of a discourse plan.

Figure 5 shows the representation of an output discourse plan corresponding to the example in Figure 4. In this figure, the proposition *the ambulance went to NYC five weekends* is represented by node 1 and objects *o*₁ and *o*₂.

Table 2. Examples of temporal predicates to describe the content of propositions.

Predicate	Description
<i>date</i> (x, y)	The date of the event x is y . For example, <i>date</i> (x_3 , XXXX-14-WXX-06TMO) represents Saturday 14 in the morning.
<i>duration</i> (x, y)	The duration of the event x is y . For example, <i>duration</i> (x_3 , PT15M) represents a duration of 15 minutes.
<i>time_interval</i> (x, y, z)	The event x starts at y and ends at z (y, z are dates). For example, <i>time_interval</i> (x_3 , 2003-01-02T16:00, 2003-01-03T08:00).
<i>periodicity</i> (x, y, z, u)	The event x happens a set of times, where y (integer) is the number of times, z is the time unit (D day, M month, etc.), u is the periodicity as a duration (ISO 8601). For example, <i>periodicity</i> (x_3 , 1, D, P2W) represents one day every two weeks.
<i>times</i> (x, y, z)	The event x happens a set of times, where y (integer or quantifier) is the number of times, and z is the time unit (D for day, M for month, etc.). For example, <i>times</i> (x_3 , 30, D) represents 30 days and <i>times</i> (x_3 , few, Y) represents a few years.
<i>dates</i> (x, y, z)	The event x happens a number of dates, where y (integer) is the number and z is the date (date with ISO 8601 representation). For example, <i>dates</i> (x_3 , 5, XXXX-XX-WE) represents five weekends.

To describe the content of propositions, we use a flat representation similar to that used in PropBank. We distinguish between three elements in this representation: general, temporal and geospatial. General elements identify the predicate-argument structure of the sentence. In node 1 of our example, *a0*(x_1, o_1) represents the subject (*the ambulance*) and *pred*(x_1, go) represents the verb. Because in this work we focus on descriptions of geospatial and temporal information, our representation for these types of information is more refined than that used in PropBank. Table 2 shows examples of predicates for temporal representation. For the dates and repetitions we follow the standard defined by ISO 8601 (which concerns formats for representing dates) and also markup languages for temporal information, in particular TimeML²⁷ and TIDES.⁶ We also use specific predicates for geospatial information like *place*(x, y), and *destination*(x, y). For these predicates we follow the markup language for spatial information SpatialML.¹³

3.2. The discourse planner

The goal of the discourse planner is to generate a discourse plan that summarizes the input location trace data. Our discourse planner was conceived as a knowledge-based hierarchical planner that constructs the discourse plan through a search directed by hypotheses of partial discourse strategies that are progressively refined until the final discourse plan is generated.³⁶ The knowledge base of the planner contains discourse patterns that correspond to partial discourse strategies at different levels of abstraction. This technical solution is fairly flexible: it permits the discourse planner to consider different types of strategy, and it can be easily extended to include new strategies.

To construct the discourse patterns currently in the knowledge base, we analyzed both human-authored geospatial/temporal summaries, and other systems that perform data-to-text summarization. Currently, the knowledge base includes strategies to construct discourses that start with a general summary statement and continue, for example, with a

list of events organized by location (e.g. visits to a place, trips between places), a list of events organized by time (e.g. a chronological sequence of trips), a small number of representative examples of events, counts of events (e.g. counts of visits to a place), etc. The knowledge base also includes strategies to combine temporal expressions about time and calendars (for example, common dates and periodicity of a set of events).

The discourse planner itself uses a modified version of HTN (Hierarchical Task Network) planning.⁸ One of the advantages of this approach is that the knowledge base can be extended to generate not only text but also multimedia presentations (as is done e.g. by Molina and Flores¹⁸). A HTN planner includes planning-tasks, planning-methods, preconditions and operators. In our approach, each planning-task corresponds to a communication goal (e.g., to inform about the details of a relevant event) and each planning-method corresponds to a discourse pattern. A pattern is defined with a set of communication sub-goals that correspond to other planning-tasks and a discourse defined by rhetorical relations. An operator corresponds to a pattern that defines either a single proposition or a set of propositions related by rhetorical relations. Thus, the whole discourse strategy comprises several levels of communication goals and candidate discourse patterns.

Figure 6 shows extracts from our knowledge base of discourse patterns. The first extract illustrates how the method relates communication goals to discourse patterns. In general, several different patterns can be associated with the same goal. In this example, the goal is to elaborate the information of a trip. The discourse pattern associated with this goal is called *chronological elaboration*. Conditions on the relationship between a goal and a discourse pattern may include abstraction goals (besides other conditions about the characteristics of the trips). In this example, the predicate *places_chronological*($x, 7, y$) represents that the trip x can be described with a maximum number of 7 places and y is the set of places. These abstraction goals are

GOAL: <i>elaborate_trip</i> (x) PATTERN: <i>chronological_elaboration</i> (x, y) CONDITIONS: $trip(x) \wedge places_chronological(x, 7, y) \wedge (y > 1)$
GOAL: <i>inform_visit</i> (x, y) PATTERN: <i>one_visit</i> (x, y) CONDITIONS: $trip(x) \wedge number_of_visits(x, y, n) \wedge (n = 1)$
PATTERN: <i>chronological_elaboration</i> (x, y) SUBGOALS: $\{goal(inform_visit(x, z), p1), goal(list_chronological(y), p2)\}$ DISCOURSE: $relation(elaboration_general_specific, nucleus(\{p1\}), satellite(\{p2\}))$ CONDITIONS: $trip(x) \wedge places_of_trip(x, 1, \{z\})$
PATTERN: <i>one_visit</i> (x, y) DISCOURSE: $proposition(\{event(e), place(e, y), pred(e, go), duration(e, z), a0(e, v)\})$ CONDITIONS: $trip(x) \wedge visit_duration(x, y, z) \wedge agent(x, v) \wedge new_event(e)$
...

Fig. 6. Partial example of knowledge representation for discourse patterns.

carried out by the abstraction component (see more details in the next section). They are invoked backwards when the discourse planner checks the conditions of a candidate goal or a candidate pattern.

The third extract in Figure 6 shows the description of the discourse pattern *chronological-elaboration*(x, y). This pattern includes a list of subgoals. Each subgoal is identified with the predicate *goal*(g, p), where g is the goal and p is a variable that identifies the partial discourse generated for this goal. This pattern uses a rhetorical relation (the *elaboration_general_specific* relation) and the variables associated with it establish the roles for nucleus and satellite in the relation.

The last extract shown in Figure 6 is an example of an operator, i.e., a pattern that does not define communication subgoals. The pattern defines the content of a proposition. In this case, the goal is to inform about a place that was visited only once.

To construct a discourse plan our discourse planner searches the space of possible discourses defined by the discourse patterns (Figure 7). During the search, the discourse planner repeatedly (a) selects a candidate discourse pattern for the current goal, (b) gathers additional abstracted information according to the conditions, and (c) refines

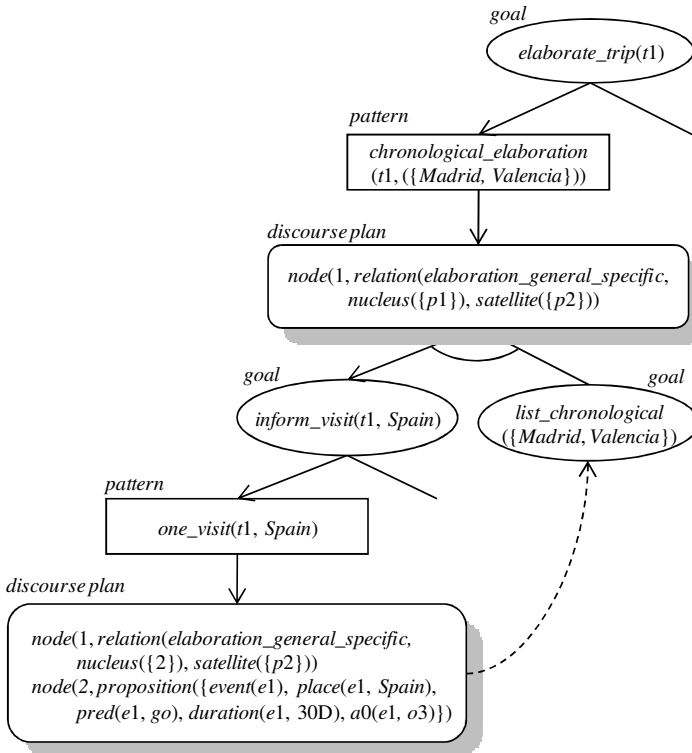


Fig. 7. Example of partial search tree developed during the discourse planning process. This example uses the knowledge base shown in Figure 6. In the figure, ellipses represent communication goals, rectangles represent discourse patterns and rounded rectangles represent the (partial) discourse plan as it is being constructed.

the communication goal with the sub-goals. Through this process, the discourse plan is dynamically constructed.

3.3. Geospatial and temporal abstraction

The abstraction component addresses the following problem: given a set of input traces, there are many ways in which they can be clustered. Furthermore, each clustering implicitly defines an abstraction over the input. However, not all abstractions are equally good. Some are too fine-grained, some too coarse-grained, and some cannot be easily expressed in natural language. How can we choose the right abstraction for a particular presentation? For example, consider the ship route shown in Figure 1. This route spans 50 days and almost 2 months. It spans 521 latitude/longitude pairs, 6 countries, 4 seas/gulfs/ rivers and 1 ocean. In a particular summary, should we choose to present these traces using a geospatial or temporal abstraction, and at what level?

One way to use abstraction would be to do a total hierarchical clustering of the input data along each dimension of interest (here, geospatial and temporal dimensions). For example, we can do this using hierarchical agglomerative clustering and then prune clusterings that cannot be easily expressed using natural language. We could then pass the results of the clustering as input to the discourse planner. However, this approach

Table 3. Examples of predicates to describe geospatial and temporal abstraction goals.

Abstraction goals	Description
<i>places_of_trip</i> (x, y, z)	Given a trip x and a maximum number of places y , this predicate generates a set z of n places which includes all the places visited. The level of abstraction for each place is selected to satisfy $n \leq y$.
<i>places_chronological</i> (x, y, z)	Given a trip x and a maximum number of places y , this predicate generates an ordered set z of n places which includes all the places visited in chronological order. The level of abstraction for each place is selected to satisfy $n \leq y$.
<i>places_by_duration</i> (x, y, z)	Given a trip x and a maximum number of places y , this predicate generates an ordered set z of n places which includes all the places visited sorted by duration. The level of abstraction for each place is selected to satisfy $n \leq y$.
<i>places_by_visits</i> (x, y, z)	Given a trip x and a maximum number of places y , this predicate generates an ordered set z of n places which includes all the places visited sorted by number of visits. The level of abstraction for each place is selected to satisfy $n \leq y$.
<i>periodicity_of_visits</i> (x, y, z, u, v)	Given a trip x and a place y that was visited several times, this predicate finds out the periodicity of the visits, described by z (integer) number of times, u is the time unit (D for day, M for month, etc.), and v is the periodicity represented as a duration (ISO 8601).
<i>date_of_visits</i> (x, y, z)	Given a trip x and a place y that could be visited several times, this predicate generates the value z that represents the common date of the visits.
<i>number_of_visits</i> (x, y, z)	Given a trip x and a place y , this predicate generates z , the number of visits to the place y .

has two big drawbacks: (1) it is slow (hierarchical agglomerative clustering is $O(n^2)$); and (2) it involves a lot of unnecessary work (since for each proposition in a summary we may only care about one dimension, or only about a subset of the input data). Consequently, we do abstraction only on an as-needed basis during discourse planning.

We defined a set of geospatial and temporal abstraction goals. Table 3 shows examples of these goals; for instance, the goal *places_of_trip*(x, y, z) represents that the trip x can be described using a set z of n places that includes all the places visited, taking into account $n \leq y$. When the discourse planner needs an abstraction over the input data to use a particular discourse pattern, it calls the abstraction component and provides a subset of the input data and an abstraction goal. The abstraction component attempts to satisfy the abstraction goal on the input data (for this purpose, we use hierarchical agglomerative clustering on the dimension corresponding to the abstraction goal). If it succeeds, it returns a partitioning over the input data corresponding to the abstraction.

Geospatial abstraction uses a geographical database providing information based on a geographical ontology that relates toponyms of places. The following attributes are considered for geospatial abstraction: *latitude*, *longitude*, *landmark*, *address*, *street*, *road*, *neighborhood*, *town/city*, *region*, *state*, *country*, and *continent*. For our traces, we obtain this information from *geonames.org*.

Temporal abstraction goals include goals relating to time, and to periods/repetitions of temporal sequences. For this purpose, we use a knowledge base that includes information about time and calendars. These do not fall in a strict taxonomy; the set we use is shown in Table 4.

We illustrate the abstraction procedure using two examples. In order to implement the abstraction procedure for the predicate *periodicity_of_visits*(x, y, z, u, v) we use rapid approximate hierarchical agglomerative clustering.¹¹ The features include measures of duration in hours, days, weeks and so on. This procedure generates descriptions such as *3 hours every 2 days*. To implement the abstraction procedure for the predicate *date_of_visits*(x, y, z), we also use hierarchical agglomerative clustering. The features here are measures of a point in time, including the hour, the day, the week of an event, and so on. This procedure produces descriptions such as *every summer* or *many weekends*.

3.4. Text generation

The *text generator* produces a natural language text for an input discourse plan. Text generation is done by two subtasks, *sentence planning* and *surface realization*, that correspond to the last steps followed by most natural language generation systems.^{23, 25}

The sentence planner decides how pieces of content should be organized into sentences, and how the relationships between them should be expressed using discourse cues (e.g. *for example*, *and*). For the sentence planner we use a modified version of the publicly available SPaRKY sentence planner.³² SPaRKY takes as input a discourse plan (a tree with rhetorical relations on the internal nodes and a proposition representing a text

Table 4. Examples of attributes used for temporal abstraction.

Attribute	Values
Julian day	Number of days since 12:00 on January 1st of year -4712 (real number)
Epoch day	Number of seconds since 00:00 on January 1, 1970.
Minute	{ 1, ..., 59 }
Quarter of an hour	{ first, second, third, fourth }
Half an hour	{ first, second }
Hour	{ 1, ..., 24 }
Part of day	{ morning, afternoon, evening, night }
Third of day	{ early, middle, late }
One-day holiday	{ Thanksgiving, Christmas, Columbus Day, ... }
Day	{ 1, ..., 31 }
Day of week	{ Monday, ..., Sunday }
Few-day holiday	{ Christmas, Easter, ... }
Part of week	{ weekday, weekend }
Third of week	{ early, middle, late }
Number of week	{ 1, 2, ... }. Number of week in the year.
Third of month	{ early, middle, late }
Month	{ January, ..., December }
Season	{ winter, spring, summer, fall }
Quarter	{ first, second, third, fourth }
Third of year	{ early, middle, late }
Semester	{ first, second }
Year	{ ..., 1995, 1996, ... }
Financial year	{ ..., 1995, 1996, ... }
Third of decade	{ early, middle, late }
Decade	{ ..., 1980's, 1990's, ... }
Third of century	{ early, middle, late }
Century	{ ..., 18, 19, 20, ... }

span on each leaf), and outputs one or more sentence plans (each a tree with discourse cues and/or punctuation on the internal nodes).

SPaRky is a two-stage sentence planner. First, possible sentence plans are constructed through a sequence of decisions made using only local information about single nodes in the discourse plan. Second, the possible sentence plans are ranked using a user or domain-specific sentence plan ranker that evaluates the global quality of each sentence plan.³⁴

Sentence plan construction in SPaRky involves three tasks: span ordering, sentence aggregation (deciding whether to realize a pair of propositions as a single clause, a single sentence, or two sentences), and discourse cue selection. In our version of SPaRky, for sentence plan construction we use sentence planning rules automatically extracted from the RST-DT corpus¹ as well as some hand-written rules (see details about the extraction procedure in a previous publication³¹). For sentence plan ranking, we use an n-gram

language model trained on the APW section of the English Gigaword corpus.⁹ Examples of sentence plan rules are shown in Figure 7.

The surface realizer generates text from the abstract representations for each sentence. For the surface realizer we currently use SimpleNLG.⁷ SimpleNLG is a rule-based surface realizer. We perform a straightforward procedure to automatically transform our flat semantic representation for propositions into SimpleNLG input. When the sentence planner decides that two or more propositions should be aggregated into a single clause, we merge co-referential arguments and identical predicates in the input propositions. When the sentence planner decides that two or more propositions should be aggregated into a single sentence (separated by a comma or semi-colon), we realize each proposition separately and then insert the desired punctuation.

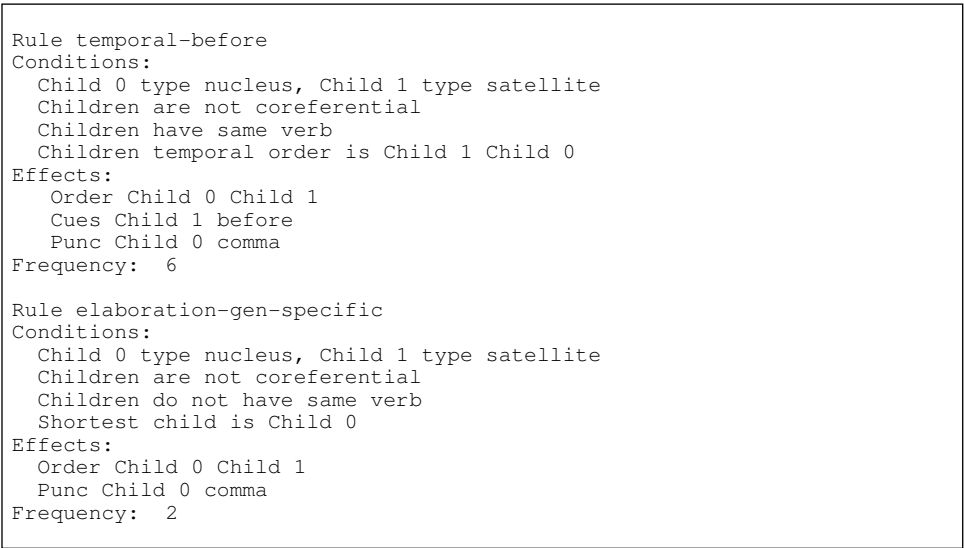


Fig. 7. Glosses of example sentence plan rules.

4. Evaluation

We have constructed an initial implementation of our summarization method and performed a preliminary evaluation of it. The discourse planner is implemented in Prolog. The text generator is implemented in Java with a modified version of SPaRKY³² for the sentence planner and adaptation of SimpleNLG⁷ for the surface realizer (the rules for the surface realizer are stored in XML).

Our initial implementation also includes a multimedia presentation module that provides context for the text summaries. Figure 8 shows an example of our design for a multimedia composition that presents summaries about trips. In the figure, on the left hand side, a text description summarizes the trip. The geographical presentation includes

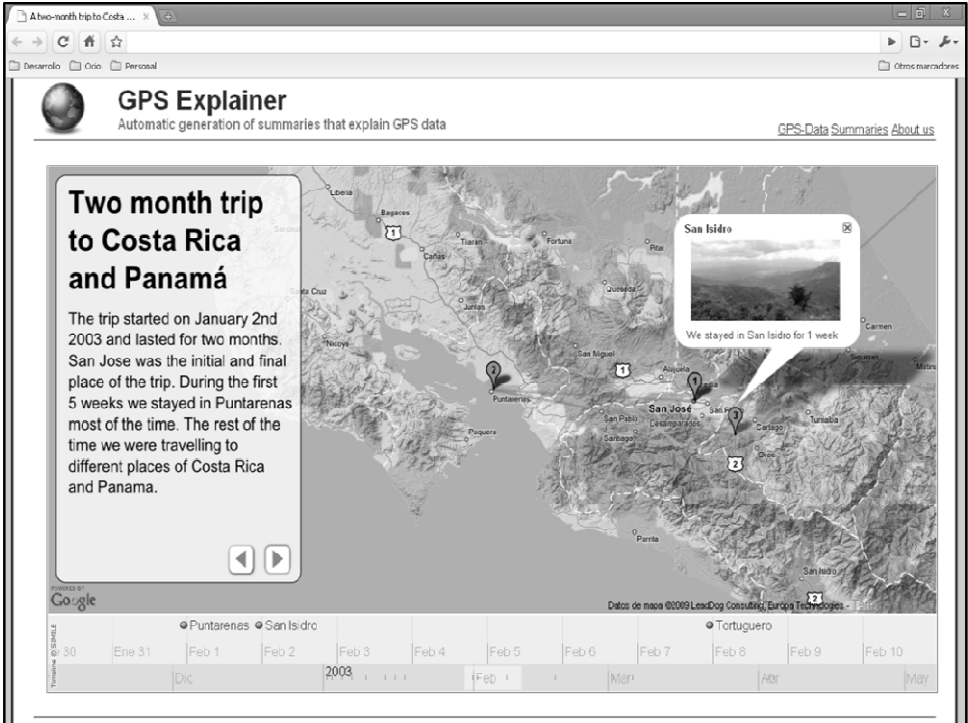


Fig. 8. Example of presentation of geographical and temporal information.

description points for specific places on the map; when the user clicks on one he or she is presented with a specific summary. The overall presentation also includes a temporal bar at the bottom of the map, and the user can go forward and backward in time using the left and right arrows at the bottom of the text description. For this module we used tools from the *SIMILE Project*.¹⁶

To evaluate this implementation, first, we carried out an evaluation of the coverage and discriminative power of our sentence planning rules (see details about this in a previous publication³¹). We have also conducted an evaluation of the quality of the natural language summaries generated by our prototype implementation. We applied a procedure based on the comparison of (1) *gold standard* summaries, in this case summaries of the input data created by two human annotators and (2) the output (text in natural language) generated automatically by our prototype system. For this purpose we first selected a set of 20 descriptions about trips from the plan summaries produced by participants in the Monroe dialog collection.³⁰ Each example describes one or more trips made by a single vehicle in and around Rochester, NY. For each example, the corresponding raw data was manually generated (we used as input data time-ordered list of pairs <toponym, time-interval> instead of raw GPS data, so the evaluation did not include data pre-processing). The raw data was processed by our method to automatically generate new text descriptions. For each input our system produced up to 100 summaries,

which were ranked using the sentence planning rule weights and an n-gram language model. We then compared the generated summaries with two human-authored summaries for each set of raw data.

The average length of the highest-ranked generated summaries was 39.7 words (with 3 punctuation marks indicating clause boundaries), as compared with 49.1 words (and 4.1 punctuation marks) for summaries generated by one human and 47.7 words (and 3.25 punctuation marks) for summaries generated by the other.

We conducted an automatic evaluation of the generated summaries. We took the highest-ranked summaries for each input and evaluated them against the two human-authored summaries using the widely-used NIST²⁰ and BLEU²¹ evaluation metrics (for a brief summary of NIST and BLEU see Appendix B). We achieved overall NIST score of 5.444, and overall BLEU score of 0.492, indicating a medium level of similarity between the human-authored and generated summaries. As illustrated in Figure 9, the human-authored summaries also differ quite a bit. Automatic evaluation scores are in the range of 5.3 to 5.6 (NIST) and 0.47 to 0.51 (BLEU) for the ten most highly-ranked summaries. This is not surprising, as the mostly highly-ranked summaries differ mainly in sentence order and choice of cue words, which carry nuances that cannot be evaluated using automatic metrics.

For this reason, we also conducted a human evaluation of the highest-ranked generated summaries. Two human labelers separately rated each summary on a scale from 0 (worst) to 1 (best) according to each of the following desiderata: (1) correctness, i.e., the generated text does not contradict the human-authored summaries, (2) grammaticality and naturalness, i.e., the formulation of the generated text is fluent and idiomatic English, (3) completeness, i.e. the generated text covers everything said in the human-authored summaries, and (4) equivalence, i.e., the generated text is equal to or better than the human-authored summaries (no word would be changed). The summaries obtained an average rating of 0.96 for correctness, 0.96 for naturalness and grammaticality, 0.91 for completeness, and 0.87 for equivalence. The evaluation showed that the method generates acceptable text descriptions, especially for certain spatial aggregations and chronological descriptions of trips. These results correspond to an improvement compared to a reported evaluation of a previous version of our method.¹⁹ The improvement was achieved by refining the content of knowledge bases (for example, adding new rules to the sentence planner for titles and pronouns).

Our evaluation also included a number of comparison tests with other more complex examples based on texts found in public web sites. In general, the method generated correct texts, and periodic patterns were adequately described. Besides this, the method presented some limitations that could be solved by including additional knowledge-based components. Some of these limitations are the following:

- Our method describes geographical places with names of locations using different spatial granularities. However the current version of our method is not able to present richer geographical descriptions about relative positions based on distance,

<i>Human 1</i>	<p>2 places visited in Rochester</p> <p>The second road crew visited 2 places in Rochester in the afternoon of April 1 2007. First, the crew went from the Greater Rochester Airport to the Elmwood Bridge for 3 hours. Then they went to the intersection of Thurston Road and Brooks Avenue for 2 hours.</p>
<i>Human 2</i>	<p>The second road crew takes a two-stop trip</p> <p>The second road crew left Greater Rochester Airport at 12:00 p.m. It arrived at Elmwood Bridge at 12:10 p.m. and stayed for three hours. Then it went to the intersection of Thurston Road and Brooks Avenue, arriving at 3:15 p.m. and staying for two hours.</p>
<i>Generated text</i>	<p>Visit to 2 places in Rochester</p> <p>The second road crew visited 2 places in Rochester for 5 hours. It visited Elmwood Bridge for 3 hours. Then it visited the intersection of Thurston Road and Brooks Avenue for 2 hours.</p>
<i>Human 1</i>	<p>2 trips in Rochester</p> <p>The ambulance made 2 trips in Rochester from the Strong Memorial Hospital during about 3 hours in May 1 2007. In the first trip, the ambulance went to the intersection of East Avenue and I-590 for 15 minutes. In the second trip, the ambulance went to the intersection of NY-252 and East River Road for 15 minutes.</p>
<i>Human 2</i>	<p>The ambulance takes two trips from Strong Memorial Hospital</p> <p>The ambulance took two round trips from Strong Memorial Hospital. The first trip, to the intersection of East Avenue and I-590, took 35 minutes. The second trip, to the intersection of NY-252 and East River Road, took only 25 minutes.</p>
<i>Generated text</i>	<p>2 trips in Rochester</p> <p>The origin of every trip was at Strong Memorial Hospital. The ambulance made 2 trips on May 1 2007 in Rochester. It went at 12 o'clock to the intersection of East Avenue and I590 for 15 minutes. Then it went at 13 o'clock to the intersection of NY252 and East River Road for 15 minutes.</p>
<i>Human 1</i>	<p>4 places visited in Rochester</p> <p>The first bus went to 4 places in Rochester in January 1, 2009 from 12:00 to 14:55. It went three times to Irondequoit Police Station, twice to Seabreeze Amusement Park and once to Lake Shore Country Club and Crescent Beach.</p>
<i>Human 2</i>	<p>The first bus takes three trips</p> <p>The first bus took three trips to Irondequoit Police Station. On the first trip it went from Pittsford Fire Station via Lake Shore Country Club and took 1 hour total. On the second trip it went to Seabreeze Amusement Park and took 35 minutes total. On the final trip it went to Seabreeze Amusement Park and Crescent Beach, and took 50 minutes total.</p>
<i>Generated text</i>	<p>Visit to 4 places in Rochester</p> <p>The first bus visited 4 places in Rochester for 2 hours. It went to Lake Shore Country Club. It went 3 times to Irondequoit Police Station. It went 2 times to Seabreeze Amusement Park. It went to Crescent Beach.</p>

Fig. 9. Examples of text automatically generated for the Monroe Corpus.

orientation, etc., or to take into account the knowledge of the user about the geographical area. An example of a location description our system is not currently capable of producing is *Bajamar, which is 25 miles from Puntarenas*. As part of our future work we plan to extend our method with more detailed spatial knowledge and user models so that we can generate these types of descriptions.

- Concerning temporal abstractions, our representation was designed for typical temporal expressions including temporal patterns. However, this representation is not exhaustive, i.e. there are certain types of sentences in natural language that the current version of our method is not yet able to generate. For example, the following expressions cannot currently be generated: *the last Saturday of every month* or *from Monday to Thursday some weeks in 2003*. However, this could be solved by including additional temporal predicates; for example, a predicate like *set-of-times-intervals*(e , x , y , z) where e is the event, x (a integer or a name of a quantifier) is the number of times, y is an initial date and z is a final date.
- Finally, some of the attributes used for temporal abstraction depend also on geographical area. For example, the season depends on the hemisphere of the Earth and the part of the day (morning, afternoon, night) depends on the part of the year and the geographical area. In our current implementation, this is not considered but it could be easily solved in future versions by adding additional components or knowledge bases.

5. Related Work

Our summarization work involves abstraction and modeling of geospatial/temporal data, and summarization systems. In the area of temporal abstraction, different methods have been proposed; for example, methods for reasoning about temporal granularities with algebraic or logical approaches⁴ or methods that follow a knowledge-based approach for temporal abstraction.^{2,28} Our method takes a knowledge based approach but it uses a different representation for domain knowledge (events, abstraction goals, patterns, etc.) that considers not only the temporal dimension but also geospatial references.

Our method is also related to summarization systems. In this research area, there are text-to-text summarization systems that generate text descriptions that summarize source text (for example, by sentence compression³). Our work is closer to data-to-text summarization systems^{10,15,26,37} that generate text descriptions that summarize non-linguistic data. As a main difference, these proposals have been designed for particular domain tasks, while our method is designed for a particular type of data that spans many tasks (geospatial/temporal data). Therefore, the components can be used with other domain- or task-specific components (e.g. for recommendations, decision support, hypothesis exploration). This goal is facilitated by our modular knowledge-based design that allows reuse not only of specific search procedures but also of the representations and the content of our knowledge bases (wholly or partially).

There are several systems that perform data-to-text summarization of geographical data; these include Coral⁵ and RoadSafe.³³ Coral gives travel directions (based on spatial data, but without temporal references) and RoadSafe explains how weather conditions affect certain spatial areas (without considering historical behavior). In contrast, our method provides summaries along both geospatial and temporal dimensions. On the other hand, Coral generates certain spatial descriptions related to paths (such as *follow the Main Street*) that the current version of our method is not able to generate.

6. Conclusions

In this article we have presented a method for automatic generation of text summaries over geospatial/temporal data sets. To our knowledge, this is the first method designed with the goal of summarizing data corresponding to traces of spatial movement in geographical areas. Our evaluation of a prototype implementation showed that our method can generate adequate descriptions of recurrent spatio-temporal patterns and event sequences using different scales of abstraction over space and time. Our method shares certain components of a common general architecture with some data summarization systems in other domains. However, it includes certain particular data representations for spatio-temporal information and specific algorithms for inference procedures (e.g., abstraction, discourse planning).

A focus of our initial research has been constructing shareable and reusable representations for descriptions of geospatial/temporal information, for data abstraction, discourse planning and text generation. We have adapted and combined elements of representations including PropBank, TimeML, and SpatialML to create a general representation for geospatial/temporal propositional content. We have integrated these with taxonomies of geospatial/temporal abstractions that are easy to present using natural language. We have also constructed sentence planning operators, and discourse planning strategies for describing geospatial/temporal information. These representations and models correspond to commonsense knowledge that can be reused in a variety of domains and, therefore, they can help developers to build summarization systems for other domains that share geospatial and temporal dimensions.

We are currently working to improve our prototype implementation in several ways. We are doing further development of components for sentence planning and text planning to increase robustness and naturalness of the output summaries. We are experimenting with faster and more accurate abstraction methods. We are also exploring the integration of text generation with interactive visualization tools. In addition to improving existing components of our prototype system, we plan to extend our method with additional capabilities. For example, we plan to add (1) spatial knowledge to generate descriptions about relative spatial positions, (2) user models, so that we can generate spatial descriptions based on the user's familiarity with the area, and (3) models for multiple individuals, to generate summaries about fleets of vehicles or sets of travelers.

Appendix A. Standards for Representing Spatial and Temporal Information

This appendix contains brief summaries of standards for representing spatial and temporal information that we use or borrow from in our system.

A.1. NMEA

NMEA is an international standard for data communication in many types of marine electronic device, including GPS receivers. Our system can handle and interpret input NMEA GGA messages, which consist of the time (UTC), the latitude, the longitude, the altitude and additional information (which we currently discard).

A.2. SpatialML

SpatialML is an XML-based markup language for annotating locations and spatial relationships in text.¹³ For example, the sentence *We stayed in a hotel near Madrid* might be marked up as:

```
We stayed in a<PLACE id="p1" form="NOM">hotel</PLACE>
<SIGNAL id="s1">near</SIGNAL>
<PLACE country="ES" id="p2" form="NAM">Madrid</PLACE>
<LINK source="p1" target="p2" signals="s1" linkType="NR"/>
```

A.3. ISO 8601

ISO 8601 (the latest version is from 2004) is an international standard for numeric representation of dates and times. The basic format for a date is *YYYYMMDD* (the extended format is *YYYY-MM-DD*), where *YYYY* is the year, *MM* is the month and *DD* is the day (in the Gregorian calendar); if only part of a date is known only part is used, e.g. *1984* for just the year, or *YYYY-02* for February of an unspecified year. The basic format for a time is *hh:mm:ss* where *hh* is the hour, *mm* is the minutes and *ss* is the seconds. To represent a date-time one may use *YYYYMMDDThhmmssZ* or *YYYY-MM-DDThh:mm:ss*. There are many additional complexities. For example, to specify a week in a year the format *YYYYWww* is used (e.g. *2010W01* represents the first week in 2010). To specify a duration the format *PnYnMnDTnHnMnS* is used (*P* precedes the date portion of the duration; *Y* follows the number of years, *M* the number of months, *D* the number of days. *T* precedes the time portion of the duration; *H* follows the number of hours, *M* the number of minutes and *S* the number of seconds).

A.4. TimeML

TimeML is an XML-based markup language for annotating events, time expressions, and temporal relationships in text.²⁷ For example, the sentence *We traveled for three days* might be marked up as:

```
We<EVENT eid="e1" class="OCCURRENCE">
traveled</EVENT><MAKEINSTANCE eid="e1" eventID="e1" pos="VERB"
tense="PAST" aspect="NONE" polarity="POS"/>
<SIGNAL sid="s1">for</SIGNAL><TIMEX3 tid="t1" type="DURATION"
value="P3D">three days </TIMEX3>
<TLINK eventInstanceID="e1" signalID="s1"relatedToTime="t1"
relType="DURING"/>
```

Note that TimeML uses a version of the TIMEX tag from the TIDES temporal markup language,⁶ and uses ISO8601 for the representation of dates and times in the TIMEX tag.

Appendix B. BLEU and NIST Evaluation Metrics

The BLEU and NIST evaluation metrics were both intended for evaluation of the output of machine translation systems, but have been frequently used to evaluate the output of natural language generation systems as well. The BLEU and NIST metrics are computed

over word ngrams (word sequences of length n , where n typically varies from 1 to 4) from a computer-generated output and one or more reference (gold standard) outputs. The BLEU score²¹ is a weighted average over ngram lengths of the geometric mean of ngram precision scores (counts of ngrams that appear in both the generated text and the reference texts). BLEU scores range from 0 (no match to reference) to 1 (identical to reference).

The NIST score²⁰ is also a n -gram based metric, but instead of using ngram counts, It uses information-weighted ngram counts; that is, it weights less frequent (presumably more informative) ngrams higher. NIST scores range from 0 up with no fixed maximum. Both BLEU and NIST scores also incorporate adjustments to penalize overly short and overly long generated texts.

In a recent study of automatic metrics for natural language generation systems,²⁴ the best performing system among a set of systems for generating weather forecasts achieved NIST and BLEU scores of 6.871 and 0.694 respectively. The weather forecast generation task is simpler than ours, involving a smaller vocabulary, shorter presentations, and more rigid discourse and sentence structures.

Acknowledgments

This work was partially supported by the Ministry of Science and Innovation of Spain within the VIOMATICA project (TIN2008-05837/TIN). The authors thank Enrique Parodi for the development of the multimedia presentation prototype that provides context for the text summaries.

References

1. L. Carlson, D. Marcu, and M. E. Okurowski, Building a discourse-tagged corpus in the framework of rhetorical structure theory, in *Proceedings of the SIGdial Workshop on Discourse and Dialogue* (2002).
2. S. Chakravarty and Y. Shahar, A constraint-based specification of periodic patterns in time-oriented data, in *Proceedings of the Sixth International Workshop on Temporal Representation and Reasoning* (1999).
3. J. Clarke, and M. Lapata, Global inference for sentence compression: an integer linear programming approach, *Journal of Artificial Intelligence Research*, 31, (2008) 273-318.
4. C. Combi, M. Franceschet, and A. Peron, Representing and reasoning about temporal granularities, *Journal of Logic and Computation*, vol. 14, no. 1, (2004) 51-77.
5. R. Dale, S. Geldof, and J-P. Prost, Using natural language generation in automatic route description, *Journal of Research and Practice in Information Technology*, vol. 37, no. 1, (2005) 89-105.
6. L. Ferro, L. Gerber, I. Mani, B. Sundheim, and G. Wilson, TIDES 2005 standard for the annotation of temporal expressions, MITRE, (2005), http://fofoca.mitre.org/annotation_guidelines/timex2_annotation_guidelines.html.
7. A. Gatt and E. Reiter, SimpleNLG: A realisation engine for practical applications, *Proceedings of ENLG* (2009).
8. M. Ghallab D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*, (Morgan Kaufmann, 2004).

9. D. Graff, English Gigaword, *Linguistic Data Consortium Catalog No. LDC2003T05*, (2003), <http://www ldc.upenn.edu/>.
10. J. Hunter, A. Gatt, F. Portet, E. Reiter, and S. Sripada, Using natural language generation technology to improve information flows in intensive care units, in *Proceedings of the 5th Conference on Prestigious Applications of Intelligent Systems* (2008).
11. M. Kull and J. Vilo, Fast approximate hierarchical clustering using similarity heuristics, *BioData Mining*, 1:9 (2008).
12. J. Lesbegueres, C. Sallaberry, and M. Gaio, Associating spatial patterns to text-units for summarizing geographic information, in *Proceedings of Geographic Information Retrieval* (2006).
13. I. Mani, J. Hitzeman, J. Richer, D. Harris, R. Quimby, and B. Wellner, SpatialML: Annotation scheme, corpora, and tools, in *Proceedings of the Sixth International Language Resources and Evaluation Conference* (2008).
14. W. Mann and S. Thompson, Rhetorical Structure Theory: Toward a functional theory of text organization, *Text*, vol. 8, no. 3, pp. (1988) 243-281.
15. M. T. Maybury, Generating summaries from event data, *Information Processing and Management: an International Journal*, Volume 31, Issue 5 (special issue: Summarizing Text), September (1995) 735-751.
16. S. Mazzocchi, S. Garland, and R. Lee, SIMILE: Practical metadata for the semantic web, *xm1.com*, January 26, (2005).
17. K. McCoy and M. Strube, Generating anaphoric expressions: Pronoun or definite description?, in *Proceedings of the Workshop on the Relation of Discourse/Dialogue Structure and Reference* (1999).
18. M. Molina and V. Flores, A presentation model for multimedia summaries of behavior, in *Proceedings of the International Conference on Intelligent User Interfaces* (2008).
19. M. Molina and A. Stent, Generating descriptions that summarize geospatial and temporal data, in *Proceedings of the International Conference of Tools on Artificial Intelligence*, (Newark, New Jersey, USA, 2009).
20. NIST Report, Automatic evaluation of machine translation quality using n-gram co-occurrence statistics (2002), <http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf>K.
21. K. Papineni, S. Roukos, T. Ward, and W. Zhu, BLEU: A method for automatic evaluation of machine translation in *Proc. of the Annual Meeting of the Association for Computational Linguistics* (2002).
22. J. Polifroni, G. Chung, and S. Seneff, Towards the automatic generation of mixed-initiative dialogue systems from web content, in *Proceedings of EUROSPEECH* (2003).
23. O. Rambow, S. Bangalore, and M.A. Walker, Natural language generation in dialog systems, in *Proceedings of the Human Language Technology Conference* (2001).
24. E. Reiter and A. Belz, An investigation into the validity of some metrics for automatically evaluating natural language generation systems, *Computational Linguistics*, vol. 35, no. 4, (2009) 529-558.
25. E. Reiter and R. Dale, *Building Natural Language Generation Systems* (Cambridge University Press, Cambridge, UK, 2000).
26. E. Reiter, S. Sripada, J. Hunter, J. Yu, and I. Davy, Choosing words in computer-generated weather forecasts, *Artificial Intelligence*, vol. 67, no. 1-2 (2005) 137-169.
27. A. Setzer and G. Katz, TimeML: Robust specification of event and temporal expressions in text, in *Proceedings of the Fifth International Workshop on Computational Semantics* (2003).
28. Y. Shahar, A framework for knowledge-based temporal abstraction, *Artificial Intelligence*, vol. 90, no. 1-2 (1997) 79-133.

29. S. Sripada, E. Reiter, J. Hunter, and J. Yu, Generating English summaries of time series data using the Gricean maxims, in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, (2003).
30. A. Stent, The Monroe corpus, *Technical Report 728*, Department of Computer Science, University of Rochester, March (2000).
31. A. Stent and M. Molina, Evaluating automatic extraction of rules for sentence plan construction, in *Proceedings of the SIGdial Conference on Discourse and Dialogue* (2009).
32. A. Stent, R. Prasad, and M.A. Walker, Trainable sentence planning for complex information presentations in spoken dialog systems, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (2004).
33. R. Turner, Y. Sripada, and E. Reiter, Generating approximate geographic descriptions, in *Proceedings of the European Natural Language Generation Conference* (2009).
34. M. A. Walker, A. Stent, F. Mairesse, and R. Prasad, Individual and domain adaptation in sentence planning for dialogue, *Journal of Artificial Intelligence Research*, vol. 30 (2007) 413-456.
35. N. J. Yattaw, Conceptualizing space and time: a classification of geographic movement, *Cartography and Geographic Information Science*, 26 (2) (1999) 85-98.
36. M. Young and J. D. Moore, DPOCL: A principled approach to discourse planning, in *Proceedings of the 7th International Workshop on Natural Language Generation* (1994).
37. J. Yu, E. Reiter, J. Hunter, and C. Mellish, Choosing the content of textual summaries of large time-series data sets, *Natural Language Engineering*, vol. 13, no. 1 (2007) 25-49.